Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	H8S/2600
Core Size	16-Bit
Speed	28MHz
Connectivity	I²C, IrDA, SCI, SmartCard
Peripherals	DMA, POR, PWM, WDT
Number of I/O	73
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 16x10b; D/A 4x8b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 75°C (TA)
Mounting Type	Surface Mount
Package / Case	120-TQFP
Supplier Device Package	120-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/renesas-electronics-america/df2633rte28v

2.6.3 Table of Instructions Classified by Function

Table 2.3 summarizes the instructions in each functional category. The notation used in table 2.3 is defined below.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

4.5 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4.5 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4.5 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend:
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution

5.4.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5.6 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5.4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

Bit 5	Bit 4	Bit 3	
BAMRA2	**BAMRA1**	**BAMRA0**	**Description**
0	0	0	All BARA bits are unmasked and included in break conditions (Initial value)
		1	BAA0 (lowest bit) is masked, and not included in break conditions
	1	0	BAA1 to BAA0 (lower 2 bits) are masked, and not included in break conditions
		1	BAA2 to BAA0 (lower 3 bits) are masked, and not included in break conditions
1	0	0	BAA3 to BAA0 (lower 4 bits) are masked, and not included in break conditions
		1	BAA7 to BAA0 (lower 8 bits) are masked, and not included in break conditions
	1	0	BAA11 to BAA0 (lower 12 bits) are masked, and not included in break conditions
		1	BAA15 to BAA0 (lower 16 bits) are masked, and not included in break conditions

Bits 2 and 1—Break Condition Select A (CSELA1, CSELA0): These bits selection an instruction fetch, data read, data write, or data read/write cycle as the channel A break condition.

Bit 2	Bit 1		
CSELA1	**CSELA0**	**Description**	
0	0	Instruction fetch is used as break condition	(Initial value)
	1	Data read cycle is used as break condition	
1	0	Data write cycle is used as break condition	
	1	Data read/write cycle is used as break condition	

Bit 0—Break Interrupt Enable A (BIEA): Enables or disables channel A PC break interrupts.

Bit 0		
BIEA	**Description**	
0	PC break interrupts are disabled	(Initial value)
1	PC break interrupts are enabled	

Figure 8.28 shows a transfer example in which $\overline{\text{TEND}}$ output is enabled and word-size single address mode transfer (read) is performed from external 8-bit, 2-state access space to an external device.

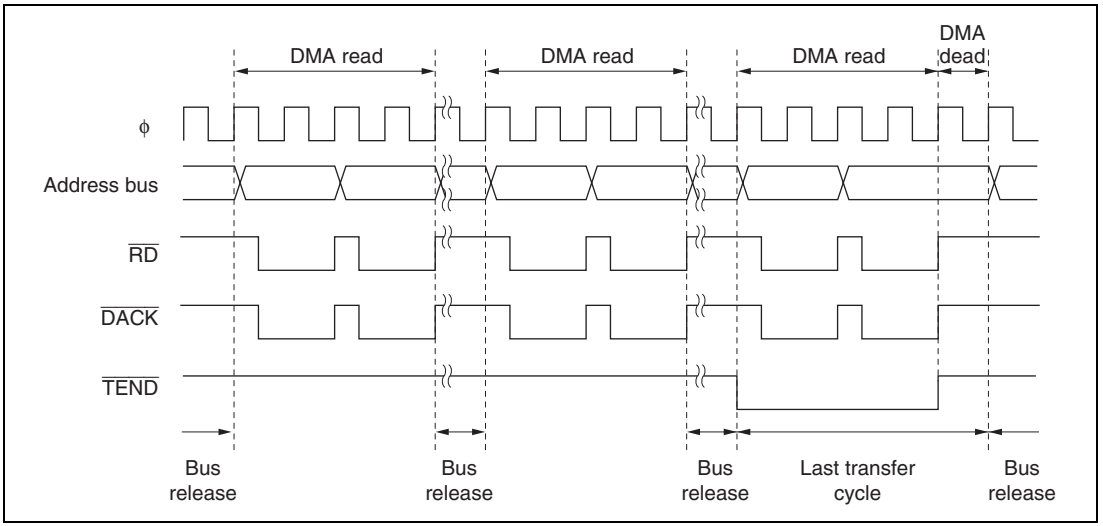


Figure 8.28 Example of Single Address Mode (Word Read) Transfer

A one-byte or one-word transfer is performed for one transfer request, and after the transfer the bus is released. While the bus is released, one or more bus cycles are inserted by the CPU or DTC.

In the transfer end cycle (the cycle in which the transfer counter reaches 0), a one-state DMA dead cycle is inserted after the DMA write cycle.

Port D pin functions in mode 7 are shown in figure 10A.14.

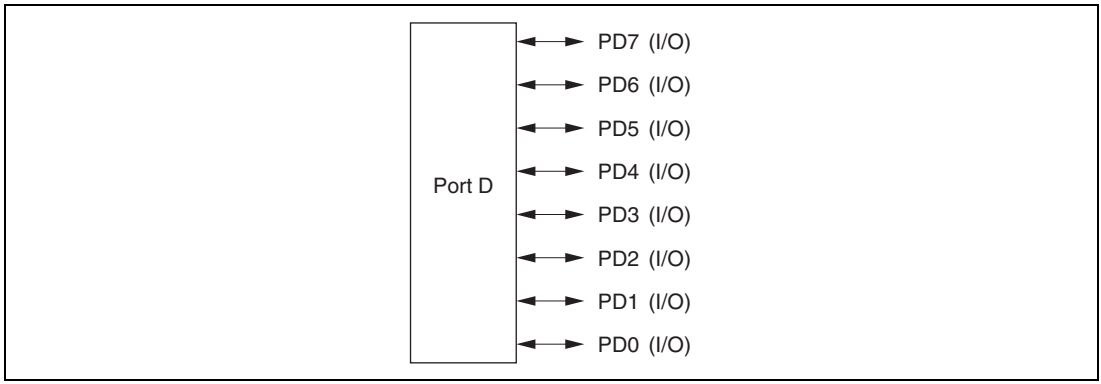


Figure 10A.14 Port D Pin Functions (Mode 7)

10A.10.4 MOS Input Pull-Up Function

Port D has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in mode 7, and can be specified as on or off on an individual bit basis.

When a PDDDR bit is cleared to 0 in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a power-on reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 10A.19 summarizes the MOS input pull-up states.

Table 10A.19 MOS Input Pull-Up States (Port D)

Modes	Power-On Reset	Hardware Standby Mode	Manual Reset	Software Standby Mode	In Other Operations
4 to 6	OFF	OFF	OFF	OFF	OFF
7			ON/OFF	ON/OFF	ON/OFF

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

10B.3 Port 3

10B.3.1 Overview

Port 3 is an 8-bit I/O port. Port 3 is a multi-purpose port for SCI I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, SCK1, TxD4, RxD4, and SCK4) and external interrupt input pins ($\overline{\text{IRQ4}}$ and $\overline{\text{IRQ5}}$). All of the port 3 pin functions have the same operating mode. The configuration for each of the port 3 pins is shown in figure 10B.2.

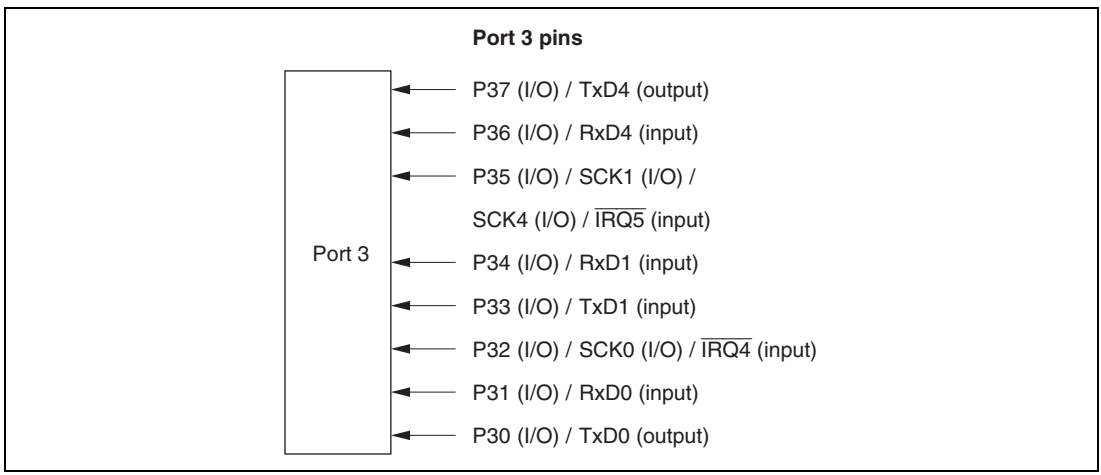


Figure 10B.2 Port 3 Pin Functions

10B.3.2 Register Configuration

Table 10B.4 shows the configuration of port 3 registers.

Table 10B.4 Port 3 Register Configuration

Name	Abbreviation	R/W	Initial Value	Address*
Port 3 data direction register	P3DDR	W	H'00	H'FE32
Port 3 data register	P3DR	R/W	H'00	H'FF02
Port 3 register	PORT3	R	Undefined	H'FFB2
Port 3 open drain control register	P3ODR	R/W	H'00	H'FE46

Note: * Lower 16 bits of the address.

Address H'FE2D

Bit	:	7	6	5	4	3	2	1	0
		NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2F

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits in NDRH (group 3) is H'FE2C and the address of the lower 4 bits (group 2) is H'FE2E. Bits 3 to 0 of address H'FE2C and bits 7 to 4 of address H'FE2E are reserved bits that cannot be modified and are always read as 1.

Address H'FE2C

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

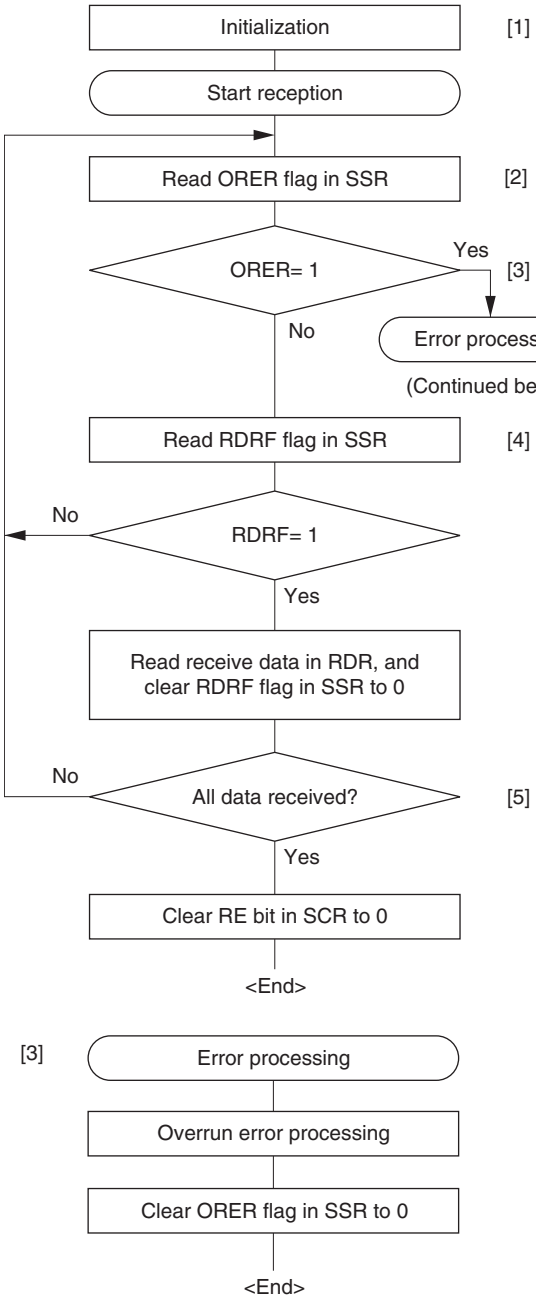
Address H'FE2E

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the address of the upper 4 bits in NDRL (group 1) is H'FE2D and the address of the lower 4 bits (group 0) is H'FE2F. Bits 3 to 0 of address H'FE2D and bits 7 to 4 of address H'FE2F are reserved bits that cannot be modified and are always read as 1. However, the H8S/2633 Group has no output pins corresponding to pulse output groups 0 and 1.

15.5.6 OVF Flag Clearing in Interval Timer Mode

When the OVF Flag setting conflicts with the OVF flag reading in interval timer mode, writing 0 to the OVF bit may not clear the flag even though the OVF bit has been read while it is 1. If there is a possibility that the OVF flag setting and reading will conflict, such as when the OVF flag is polled with the interval timer interrupt disabled, read the OVF bit while it is 1 at least twice before writing 0 to the OVF bit to clear the flag.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DMAC* or DTC* is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: * DMAC and DTC functions are not available in the H8S/2695.

Figure 16.18 Sample Serial Reception Flowchart

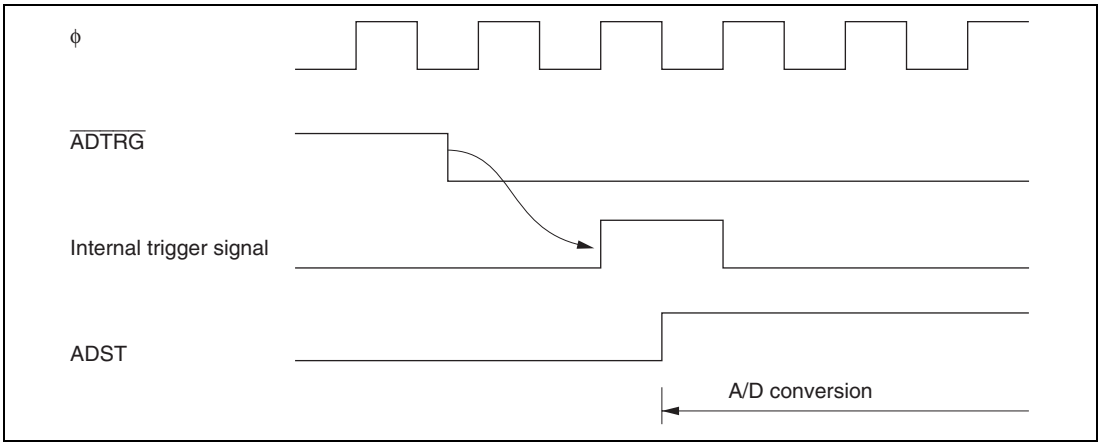


Figure 19.6 External Trigger Input Timing

19.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. ADI interrupt requests can be enabled or disabled by means of the ADIE bit in ADCSR.

The DTC* and DMAC* can be activated by an ADI interrupt. Having the converted data read by the DTC* or DMAC* in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

The A/D converter interrupt source is shown in table 19.6.

Note: * This function is not available in the H8S/2695.

Table 19.6 A/D Converter Interrupt Source

Interrupt Source	Description	DTC*, DMAC* Activation
ADI	Interrupt due to end of conversion	Possible

Note: * This function is not available in the H8S/2695.

Section 23A Clock Pulse Generator (H8S/2633, H8S/2632, H8S/2631, H8S/2633F)

23A.1 Overview

The H8S/2633 Group has a built-in clock pulse generator (CPG) that generates the system clock (φ), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator, PLL (phase-locked loop) circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit, subclock oscillator, and waveform shaping circuit. The frequency can be changed by means of the PLL circuit in the CPG. Frequency changes are performed by software by means of settings in the system clock control register (SCKCR) and low-power control register (LPWRCR).

23A.1.1 Block Diagram

Figure 23A.1 shows a block diagram of the clock pulse generator.

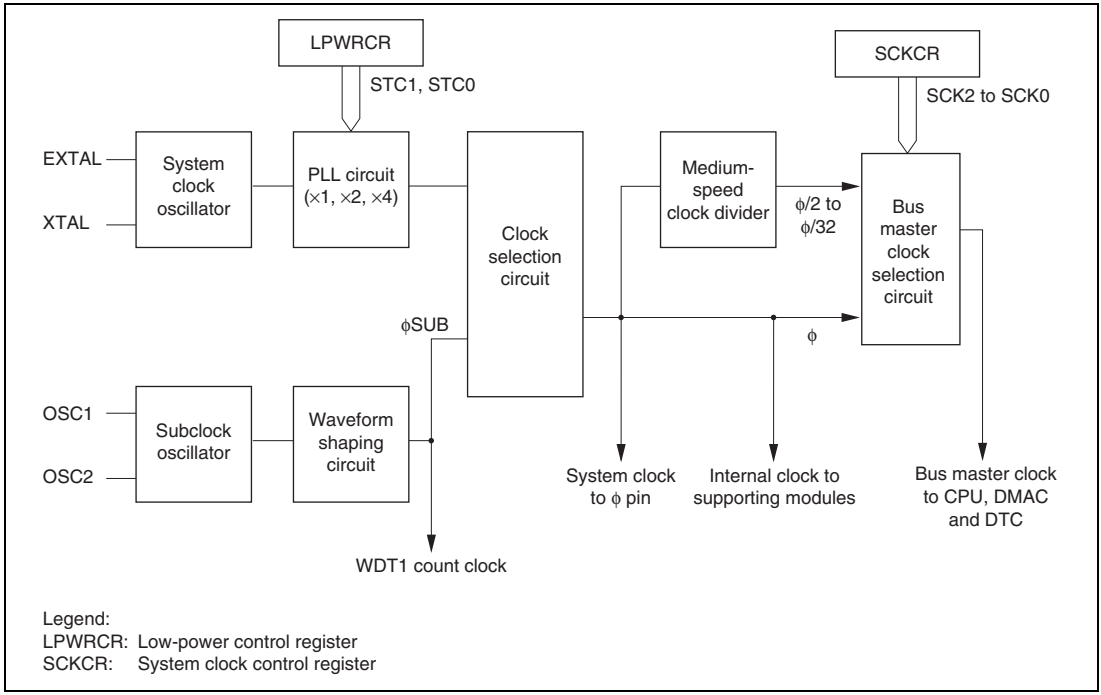


Figure 23A.1 Block Diagram of Clock Pulse Generator

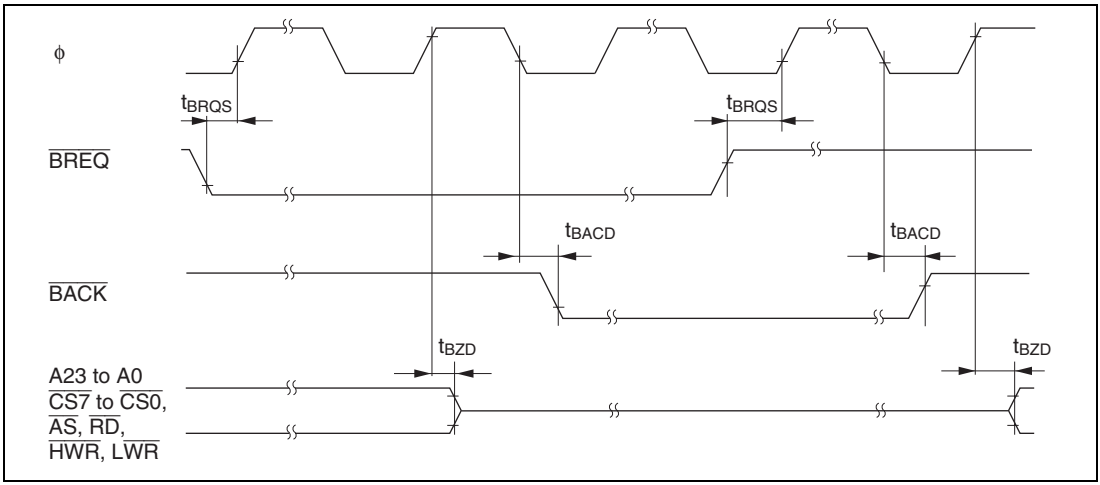


Figure 25.14 External Bus Release Timing

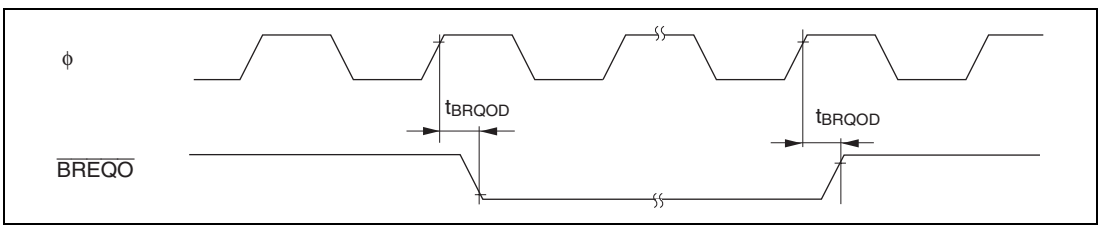


Figure 25.15 External Bus Request Output Timing

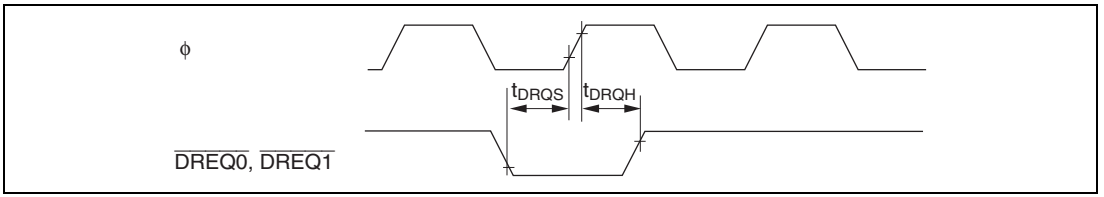


Figure 25.19 DMAC DREQ Input Timing

26.3.2 Control Signal Timing

Table 26.6 lists the control signal timing.

Table 26.6 Control Signal Timing

Conditions: PV_{CC} = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = AV_{SS} = $PLLV_{SS}$ = 0 V, 2 to 28 MHz, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item	Symbol	Min	Max	Unit	Test Conditions
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 26.4
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
$\overline{MRES}$ setup time	t_{MRESS}	250	—	ns	
$\overline{MRES}$ pulse width	t_{MRESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 26.5
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

Instruction	1	2	3	4	5	6	7	8	9
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STC EXR, @ aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STM.L(ERn–ERn+1), @–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+2), @–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+3), @–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STMAC MACH,ERd	R:W NEXT								
STMAC MACL,ERd	R:W NEXT								
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
SUB.L ERs,ERd	R:W NEXT								
SUBS #1/2/4,ERd	R:W NEXT								
SUBX #xx:8,Rd	R:W NEXT								
SUBX Rs,Rd	R:W NEXT								
TAS @ERd*8	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA					
TRAPA #x:2	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
XOR.L ERs,ERd	R:W 2nd	R:W NEXT							
XORC #xx:8,CCR	R:W NEXT								
XORC #xx:8,EXR	R:W 2nd	R:W NEXT							

SMR3—Serial Mode Register 3	**H'FDD0**	**SCI3**
SMR4—Serial Mode Register 4	**H'FDD8**	**SCI4**
SMR0—Serial Mode Register 0	**H'FF78**	**SCI0**
SMR1—Serial Mode Register 1	**H'FF80**	**SCI1**
SMR2—Serial Mode Register 2	**H'FF88**	**SCI2**

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock select 1 and 0

CKS1	CKS0	Description
0	0	φ clock
	1	φ/4 clock
1	0	φ/16 clock
	1	φ/64 clock

Multiprocessor mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop bit length

0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

Parity mode

0	Even parity*1
1	Odd parity*2

Notes: 1. When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.
2. When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.
In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Parity enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*

Note: * When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Character length

0	8-bit data
1	7-bit data*

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

Communication mode

0	Asynchronous mode
1	Clocked synchronous mode

C.4 Port 7 Block Diagram

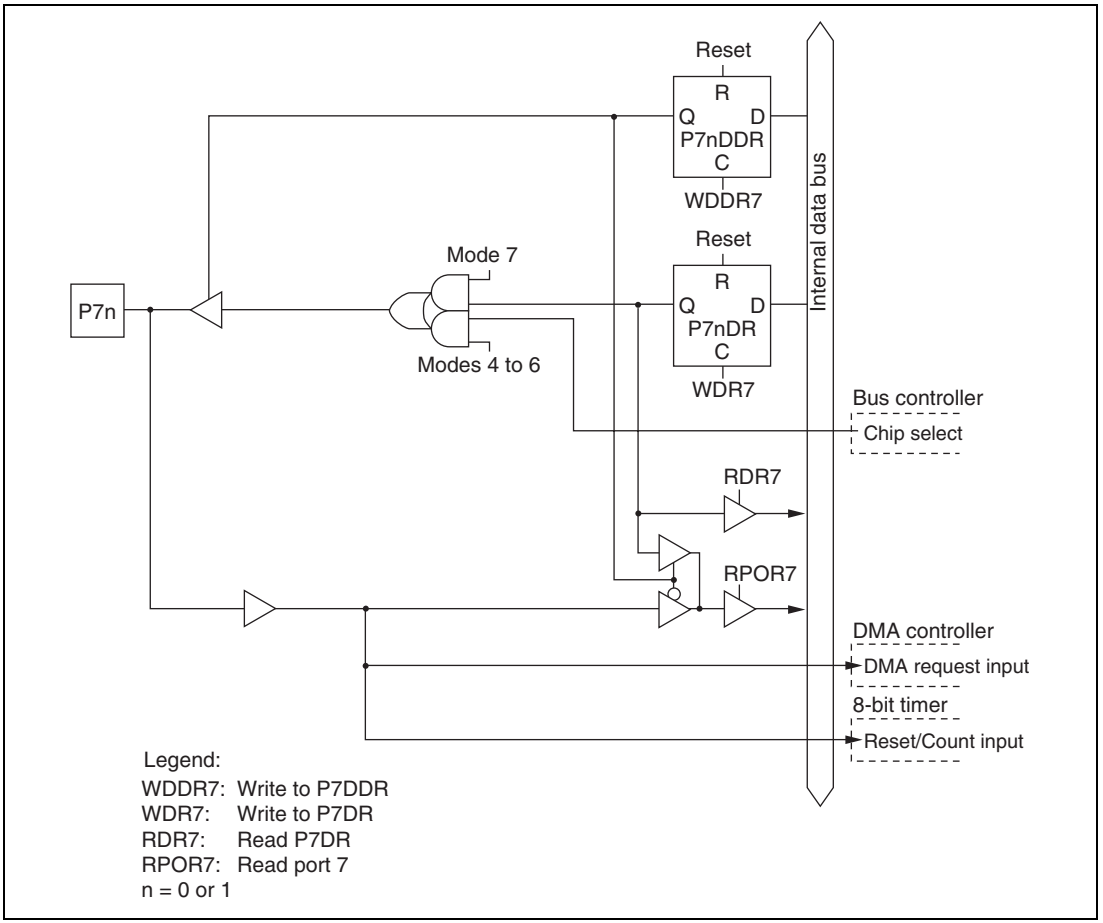


Figure C.4 (a) Port 7 Block Diagram (Pins P70 and P71)

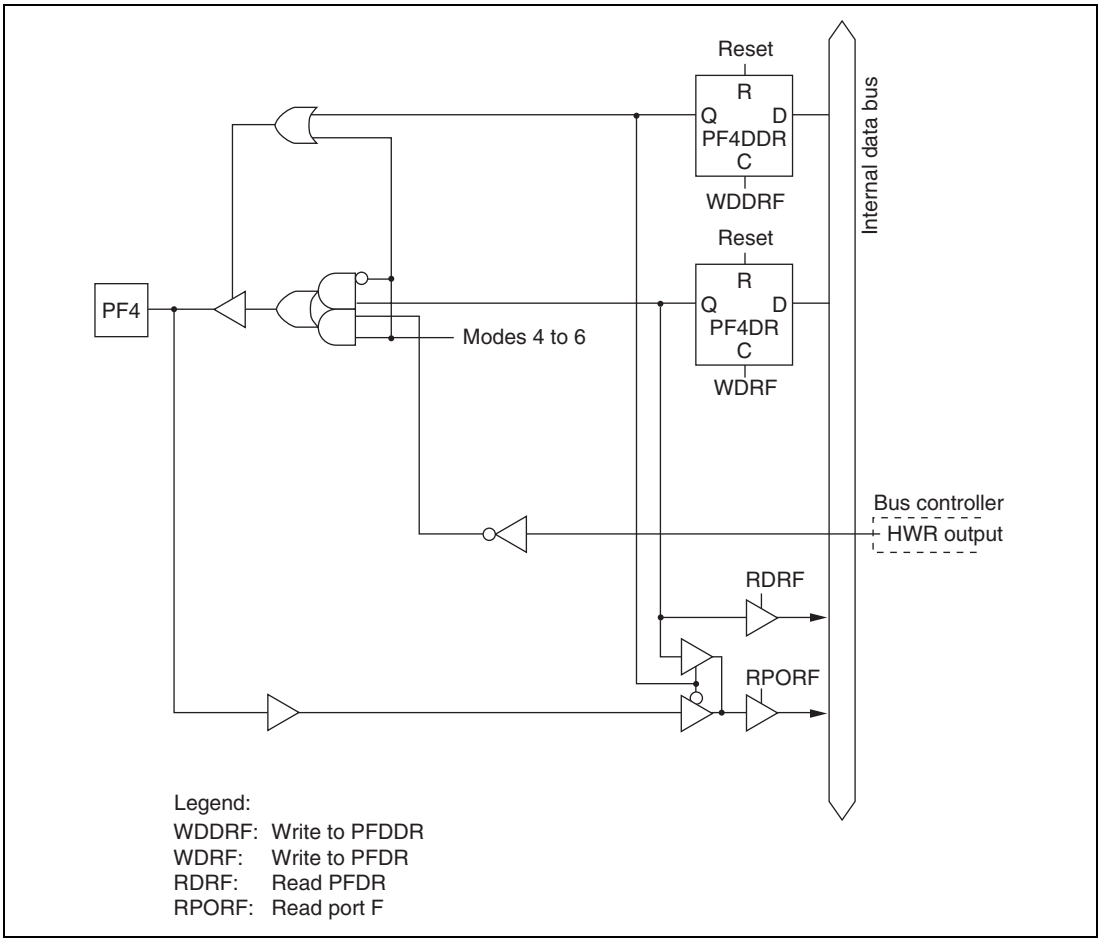


Figure C.23 (e) Port F Block Diagram (Pin PF4)

Port Name Pin Name	MCU Operating Mode	Power-On Reset	Manual Reset	Hardware Standby Mode	Software Standby Mode	Bus Release State	Program Execution State Sleep Mode
PF5/$\overline{RD}$ PF4/$\overline{HWR}$ PF3/$\overline{LWR}$/ $\overline{ADTRG}$/ $\overline{IRQ3}$	4 to 6	H	H	T	[OPE = 0] T [OPE = 1] H	T	$\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$
	7	T	kept	T	kept	kept	I/O port
PF2/$\overline{WAIT}$/ $\overline{BREQO}$	4 to 6	T	kept	T	kept	[BREQOE = 1] $\overline{BREQO}$ [WAITE = 1] T	[BREQOE = 1] $\overline{BREQO}$ [WAITE = 1] $\overline{WAIT}$
	7	T	kept	T	kept	kept	I/O port
PF1/$\overline{BACK}$	4 to 6	T	kept	T	[BRLE = 0] I/O port [BRLE = 1] H	[BRLE = 0] I/O port [BRLE = 1] L	[BRLE = 0] I/O port [BRLE = 1] $\overline{BACK}$
	7	T	kept	T	kept	kept	I/O port
PF0/$\overline{BREQ}$/ $\overline{IRQ2}$	4 to 6	T	kept	T	[BRLE = 0] kept [BRLE = 1] T	T	[BRLE = 0] I/O port [BRLE = 1] $\overline{BREQO}$
	7	T	kept	T	kept	kept	I/O port
PG4/$\overline{CS0}$	4, 5	H	kept	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] H [DDR = 0] T	T	[DDR = 0] Input port [DDR = 1] $\overline{CS0}$
	6	T					
	7	T	kept	T	kept	kept	I/O port